WEST Search History

DATE: Friday, March 26, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L12	L11 and I3	4
☐	L11	L5 and generat$ near2 stub	111
		DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ	
☐	L10	L8 and I3	7
		DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L9	L8	0
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L8	L7 and compil$	196
☐	L7	L5 and (parsing or scanning)	1024
☐	L6	L5 and (pars$ or scan$)	1850
☐	L5	instrument$ and stub$	10278
		DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ	
☐	L4	L3 or I2 or I1	3324
☐	L3	714/35,37-39.ccls.	1615
☐	L2	703/26-27.ccls.	641
☐	L1	717/124-135.ccls.	1393

END OF SEARCH HISTORY

Hit List

Search Results - Record(s) 1 through 4 of 4 returned.

☐ 1. Document ID: US 20030041288 A1

Using default format because multiple data bases are involved.

```
L12: Entry 1 of 4                        File: PGPB              Feb 27, 2003
```

```
PGPUB-DOCUMENT-NUMBER: 20030041288
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030041288 A1
```

TITLE: Method and system for dynamically invoking and/or checking conditions of a computer test program

PUBLICATION-DATE: February 27, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Kolawa, Adam	Bradbury	CA	US	
Salvador, Roman	La Jolla	CA	US	

US-CL-CURRENT: 714/38

| Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Ir |

☐ 2. Document ID: US 6611955 B1

```
L12: Entry 2 of 4                        File: USPT              Aug 26, 2003
```

```
US-PAT-NO: 6611955
DOCUMENT-IDENTIFIER: US 6611955 B1
```

TITLE: Monitoring and testing middleware based application software

DATE-ISSUED: August 26, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Logean; Xavier	Heremence			CH
Dietrich; Falk	Lausanne			CH
Hubaux; Jean-Pierre A.	Preverenges			CH
Grisouard; Sylvain	Paris			FR
Etique; Pierre-Alain	Hinterkappelen			CH
Koppenhoefer; Shawn E.	Lausanne			CH

US-CL-CURRENT: 717/128; 714/38, 719/318

ABSTRACT:

A method for on-line monitoring and testing the behavior of middleware based, distributed application software during run-time of such software is disclosed. In order to automatize the monitoring of the behavior of the application software, the monitoring method comprises: defining events capturing the behavior of the software execution, the events being based on an abstraction of the application software, the abstraction being provided by middleware; using code generating means and an instrumentation technique for automatically adding code to the implementation of the software code suitable for generating traces suitable to be sent to an observer, the information carried by the added code including information on the order of occurrence of the events and on the application software part location where each event occurs; and using a monitoring mechanism based on sending of trace reports to the observer, which ensures or takes into account the time order of the reported traces. In order to automatize testing of the behavior of the application software, the testing method comprises monitoring the behavior of the software during run-time thereof by the above mentioned method, and using a checking mechanism using the information monitored at runtime for checking whether the behavior is violating or has violated predefined properties or constraints.

16 Claims, 2 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 2

Full	Title	Citation	Front	Review	Classification	Date	Reference				Claims	KWIC	Draw Desc	Ir

---

☐ 3. Document ID: US 6546553 B1

L12: Entry 3 of 4   File: USPT   Apr 8, 2003

US-PAT-NO: 6546553
DOCUMENT-IDENTIFIER: US 6546553 B1

TITLE: Service installation on a base function and provision of a pass function with a service-free base function semantic

DATE-ISSUED: April 8, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hunt; Galen C.	Bellevue	WA		

US-CL-CURRENT: 717/174; 712/233, 712/234, 712/244, 714/38, 717/163, 717/175, 717/176, 717/177

ABSTRACT:

A base function provides a base function semantic. During service installation, an unconditional branch instruction to a service function replaces one or more instructions at the beginning of a base function. The service function provides a service semantic such as instrumentation, redirection, replacement, or extension. After service installation, a pass function includes the replaced base function instructions and an unconditional branch instruction to the logically subsequent base function instruction.

Thus, the pass function provides a service-free base function semantic. The service function calls the pass function an arbitrary number of times before and/or after executing any other service function instructions. The pass function is allocated statically or dynamically. A statically allocated pass function is callable before and/or after service installation to guarantee a service-free base function semantic. A service removal function restores a base function and conforms a pass function to the restored base function. A pass function is callable before and/or after service removal. A library of service installation functions includes functions for installing and removing a service on a base function. A library of binary editing functions includes functions for attaching service installation functions and associated data payloads to a binary file.

57 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

| Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw Desc | In |

☐ 4. Document ID: US 5313616 A

L12: Entry 4 of 4   File: USPT   May 17, 1994

US-PAT-NO: 5313616
DOCUMENT-IDENTIFIER: US 5313616 A
** See image for **Certificate of Correction** **

TITLE: Method for analyzing calls of application program by inserting monitoring routines into the executable version and redirecting calls to the monitoring routines

DATE-ISSUED: May 17, 1994

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Cline; David C.	San Jose	CA		
Silverman; Andrew P.	Los Gatos	CA		
Wymore; Farrell W.	Mountain View	CA		

US-CL-CURRENT: 717/127; 713/323, 714/35, 714/45, 717/130

ABSTRACT:

A method for verifying the conformance of an application program to a set of system rules characterized by the development of a conformance database, the performance of a static analysis of the application program to determine whether the application program is in static conformance with the conformance database and the performance of a dynamic analysis of the application program to determine whether the application program is in dynamic conformance with the conformance database. The static analysis produces a graph of the basic blocks of the application program and analyzes the graph for conformance to system rules, dead code and coverage metrics. The dynamic analysis adds a small amount of monitoring code into an executable application program which monitors the application program as it is exercised in a test harness. The monitoring code produces a log database which can be analyzed for run-time non-conformities of the application program.

11 Claims, 17 Drawing figures

CiteSeer Find: generating stub

[Documents] [Citations]

Searching for **PHRASE generating stub**.
Restrict to: Header   Title   Order by: Expected citations   Hubs   Usage   Date   Try: Amazon   B&N   Google (RI)
Google (Web)   CSB   DBLP
5 documents found. **Order: number of citations.**

Achieving Middleware Customization in a.. - Issarny, Bidan.. (1998)   (Correct)   (4 citations)
middle ware, and \Delta A set of tools **generating stub** code for the application, based on the
www.irisa.fr/EXTERNE/projet/solidor/members/../doc/ps98/iccds-b.ps.gz

Just-In-Time Stub Generation - Hof (1997)   (Correct)   (3 citations)
generation of surrogate and stub code. **Generating stub** code not in advance, but only on demand,
CORBA [COR95] or Network Objects [BiN94]is to **generate stub** and surrogate (also called skeleton code)
JustInTime **Stub Generation** Markus Hof Department of Computer Science
ftp.ssw.uni-linz.ac.at/pub/Papers/stub.ps.Z

Towards Certifying Domain-Specific Properties of Synthesized Code - Rosu, Whittle   (Correct)
ase.arc.nasa.gov/docs/../papers/VCL02/kalman-vcl02.ps

Unknown -   (Correct)
Interface Language (AIL) for automatically **generating stub** routines. Another job of the micro-kernel is
file. With this AIL code, the AIL compiler can **generate stub** codes automatically. The Bullet File Server
hwpg11.csis.hku.hk/~kkto/projects/distrep.ps.gz

IPDL - Interaction Protocols for Distributed Objects - Bokowski   (Correct)
to distributed objects, IDL files are used for **generating stub** classes for both the client and the server
helper classes would be generated by a CORBA **stub generat**or: one proxy class and one driver class for
as the target language)Note that so far, the **stub generat**or generates explicit communication operations
www.inf.fu-berlin.de/~bokowski/out/ki96ws.ps

Try your query at: Amazon   Barnes & Noble   Google (RI)   Google (Web)   CSB   DBLP

Searching for **instrumentation and stub**.
Restrict to: <u>Header</u> <u>Title</u> Order by: <u>Expected citations</u> <u>Hubs</u> <u>Usage</u> <u>Date</u> Try: <u>Amazon</u> <u>B&N</u> <u>Google (RI)</u>
<u>Google (Web)</u> <u>CSB</u> <u>DBLP</u>
35 documents found. **Order: number of citations.**

<u>Making Real-Time Reactive Systems Reliable - Marzullo, Wood (1991)</u> (Correct) (12 citations)
for distributed application management are **instrumentation** of the application, representation of the
predicate. Sensors and actuators are implemented by **stubs** that run on machines supporting the 3
instrumented registers a procedure with the sensor **stub**. The **stub** is responsible for responding to poll
ftp.cs.ucsd.edu/pub/faculty/marzullo/TR90-1155.ps.Z

<u>Ephemeral Instrumentation for Lightweight Program Profiling - Traub, Schechter, Smith (2000)</u> (Correct) (9 citations)
and S. Schechter and M.D. Smith. Ephemeral **Instrumentation** for Lightweight Program Profiling. Technical
report, Harvard University, 2000. Ephemeral **Instrumentation** for Lightweight Program Profiling Omri
www.eecs.harvard.edu/~hube/publications/pldi00.pdf

<u>The Design and Implementation of Meta - Mark Wood (1994)</u> (Correct) (5 citations)
application being instrumented. Application **instrumentation** may have an unavoidable impact on
provide facilities for carrying out control. **Instrumentation** requires explicit sensor and actu ator
with the infrastructure being provided by a **stub** that is coresident with each application
ftp.cs.ucsd.edu/pub/faculty/marzullo/isis-chapter.ps.Z

<u>On the Marginal Utility of Network Topology Measurements - Barford, Bestavros.. (2001)</u> (Correct) (3 citations)
[22] P. Barford, A. Bestavros, J. Byers, **and** M. Crovella, On the Marginal Utility of Network
www.cs.bu.edu/faculty/crovella/paper-archive/imw-marginal-utility.pdf

<u>Tools for Constructing Distributed Reactive Systems - Marzullo, Wood (1991)</u> (Correct) (3 citations)
our future plans for Meta. 2 Overview of **Instrumentation** and Control There are two steps to building
This section gives a high-level discussion of **instrumentation** and control. We introduce the basic
and Service A context is implemented with a Meta **stub**. A Meta **stub** is analogous to an RPC server **stub**
ftp.cs.ucsd.edu/pub/faculty/marzullo/TR91-1187.ps.Z

<u>Ryo: A Versatile Instruction Instrumentation Tool For Pa-Risc - Zucker, Karp (1995)</u> (Correct) (3 citations)
Ryo: A Versatile Instruction **Instrumentation** Tool For Parisc Daniel F. Zucker And Alan
gift No. 23487. RYO: a Versatile Instruction **Instrumentation** Tool for PARISC by Daniel F. Zucker and
an unconditional jump to the RYOLS library calling **stub**. This **stub** saves the state of the machine, does
umunhum.stanford.edu/tr/zucker.jan95.tr658.ps.Z

<u>Acoustic Emission Monitoring Of A Wind Turbine Blade.. - Beattie Department.. (1997)</u> (Correct) (1 citation)
time consuming for fatigue tests. Conventional **instrumentation** on these tests usually consists of strain
be obtained. Therefore several new methods of **instrumentation** aimed at acquiring failure information on
it from hitting the roof of the building. The **stub** was covered and a hydraulic actuator was attached
www.sandia.gov/Renewable_Energy/wind_energy/asme/AIAA-97-0958.pdf

<u>Pep-Ii Injection Transport Construction Status And.. - Fieguth Bloom Bulos</u> (Correct) (1 citation)
6)f%48"t Power Suppli*a93"o#9"f %stubt\1u#f7ne&u%v"09 3%xu20
www.aps.anl.gov/conferences/mirrored/www.cern.ch/accelconf/p95/ARTICLES/WAQ/WAQ15.PDF

<u>Performance of DCE RPC - Khandker Masud (1995)</u> (Correct) (1 citation)
labels imply calling a C function, when the **instrumentation** is done outside the function 1 RPC runtime
labels imply calling a C function, when the **instrumentation** is done inside the function call, i.e.when
a remote interface, control is transferred to the **stub** module for that interface in the caller's address
www.citi.umich.edu/techreports/reports/citi-tr-95-2.ps.gz

<u>Efficiently Counting Program Events with Support for On-line.. - Thomas Ball (1994)</u> (Correct) (1 citation)
programanalysis applications. We present an **instrumentation** method for efficiently counting events in a
Key Words and Phrases: controlflow graph, **instrumentation**, counting

by patching a basic block with a jump to a code **stub** rather than by rewriting the original code, the
www.bell-labs.com/user/tball/papers/eec.ps.gz

Model Based, Detailed Fault Analysis In The Cern Ps Complex .. - Beharrell Benincasa (Correct) (1 citation)
converters, RF Cavities, beam diagnostic **instrumentation**, vacuum systems, etc.For each family
level we find a series of VME crates (called Device **Stub** Controllers, DSC) housing 32bit processors of the
adwww.fnal.gov/www/icalepcs/abstracts/Postscript/wpo2.ps

Performing Replay in an OSF DCE Environment - Yuh Ming (1995) (Correct) (1 citation)
have modified the IDL compiler itself to add **instrumentation** to the **stub** code. The control required for
are captured by code in the client and server **stubs**. We have en deavoured to capture events with as
instrumenta tion. We originally modified the **stubs** by post processing the **stub** code produced by the
ccnga.uwaterloo.ca/pub/papers/Ps/conf11.ps.Z

Describing and Using Non Functional Aspects - In Component Based (2002) (Correct)
method call interception and run-time **instrumentation**. Keywords Non-Functional or Extra
component implementation choice, system **instrumentation**, debug, profiling facilities and so on. 3.
using a palette of technologies including object **(stub)** generation, method call interception and run-time
www-adele.imag.fr/Les.Publications/intConferences/AOSD2002Duc.pdf

Tree Rerooting in Distributed Garbage Collection: Implementation.. - Moreau (2000) (Correct)
in Distributed Garbage Collection: Implementation **and** Performance Evaluation. Higher-Order **and** Symbolic
www.ecs.soton.ac.uk/~lavm/papers/hosc01.ps.gz

EcoLab Documentation - Standish (1999) (Correct)
parallel.hpc.unsw.edu.au/rks/docs/ps/ecolab.ps.gz

Exploiting Software Interfaces for Performance Measurement - Konkin, Oster (Correct)
of source code ffl Automated **instrumentation** of applications, and ffl **Instrumentation** of
instrumentation of applications, and ffl **Instrumentation** of runtime libraries. 2.1.1.
Remote Procedure Call (RPC)for example, the RPC **stub** compiler could be modified to insert
ftp.cs.usask.ca/pub/discus/paper.98-2.ps.Z

An Empirical Workload Model for Driving Wide-Area - Tcp Ip Network (Correct)
layer (e.g. TCP and UDP)2 2.3. Tracing **Instrumentation** and Packet Loss Rate The UCB data was
The artificial workload model consists of a set of **stub**-network-specific arrival processes for new
new conversations between application programs, and **stub**-network-independent artificial workload model of
www.kiskeya.net/ramon/work/pubs/jinet92.pdf

An Approach to the Transparent Management.. - Villagrá.. (2002) (Correct)
An Approach to the Transparent Management **Instrumentation** of Distributed Applications Vctor A.
explains the problem of introducing management **instrumentation** in distributed application in a way that
order to be managed. The use of class wrappers and **stub instrumentation** in object-based distributed
jungla.dit.upm.es/~jlopez/publicaciones/noms02.pdf

TAU User's Guide TAU User's Guide - Version Department Of (Correct)
-12 Enabling and Disabling the **Instrumentation** -14 Using TAU with MPI-
- 15 CHAPTER 3 **Instrumentation**. 19 Automatic
. 11 TAU **Stub** Makefile- 12 Enabling
www.cs.uoregon.edu/paracomp/tau/tauprofile/docs/usersguide.ps.gz

First 20 documents Next 20

Try your query at: Amazon   Barnes & Noble   Google (RI)   Google (Web)   CSB   DBLP

CiteSeer Find: instrumentation and stub | Documents | Citations |

Searching for **instrumentation and stub**.
Restrict to: Header   Title   Order by: Expected citations   Hubs   Usage   Date   Try: Amazon   B&N   Google (RI)
Google (Web)   CSB   DBLP
35 documents found. **Order: number of citations.**

Using Shim Technology to Monitor DCE Runtime Performance - Oster, Bunt (1997)   (Correct)
of the software and hand-tuning of the **instrumentation** code in order to get the desired data. Since
DCE applications and system services: 1. Hand **instrumentation** of the application. The primary advantage of
routines) would require editing the **stub** code generated by the IDL compiler 2 It may
ftp.cs.usask.ca/pub/discus/paper.97-1.ps.Z

Wideband 3 mm SIS mixers operated with partial saturation - Engargiola And Plambeck   (Correct)
Symposium on Astronomical Telescopes and **Instrumentation** Wideband 3 mm SIS mixers operated with
a single SIS junction fed by a series inductive **stub**. The **stub** resonates out the junction capacitance,
SIS junction fed by a series inductive **stub**. The **stub** resonates out the junction capacitance, and the
bima.astro.umd.edu/memo/memo65.ps

A Java-based Remote Laboratory for Distance Learning - Sam Hsu Bassem (2000)   (Correct)
special local computer interface with a proper **instrumentation** device. This local computer is then setup as
each created for a specific task on the server. The **stub** and skeleton are the mechanism used for invoking
the different object sitting on the server. The **stubs** and skeletons of the interface are generated
www.cse.fau.edu/~bassem/Publications/Pub-33-C-ICEE2000-Taiwan.PDF

Development of a Cryogenic EOS Capability for the Z.. - Goals And.. (1998)   (Correct)
principal system components, construction and **instrumentation** of a cryogenic test facility for off-line
FY97 for interface components and screenroom **instrumentation**. We will elaborate on these accomplishments
through the thermal break and secondary hohlraum **stub** (see Figs. 4 and 9)19 12.
infoserve.sandia.gov/sand_doc/1998/980564.pdf

VisOK: A Flexible Visualization System for Distributed Java.. - Lee, Ramakrishna (2000)   (Correct)
related to the execution system. That is the **instrumentation** or tracing system which intercept op
This is very important due to the fact that **instrumentation** with tracing facility changes the program
communicate with other objects using skeleton/**stub** objects. This type of communication is clearly
parallel.kjist.ac.kr/paper/ipdps2000.ps.gz

December 1998 NASA/CR-1998-208968 - Microsoft Project-Based Planning (1998)   (Correct)
and Maintenance work package, the Model and **Instrumentation** Installation work package, and the Tunnel
ID number 37 is assigned to the `Perform **Instrumentation** Checkout'task. Additional information on
example is that the `Remove Model Sting From **Stub** Sting &Secure to Cart'task cannot start until
techreports.larc.nasa.gov/pub/techreports/larc/1998/cr/NASA-98-cr208968.ps.Z

Hierarchical Filtering-based Monitoring System for Large-scale .. - Ehab Al-Shaer   (Correct)
event reporting mechanism to facilitate the **instrumentation** process. These de sign features distinguish
monitoring system consists of four components: **Instrumentation**, Subscription Service, Event Process ing
specification which is used by the Event Reporting **Stub** in framing the notification mes sages. The ERC is
www.cs.odu.edu/~ehab/papers/pdcs97.ps.gz

Allegro: An Efficient Execution-Driven Simulator - Siegelin, O'Donnell, Finger (1995)   (Correct)
userlevel threads. A post processor adds an **instrumentation** code to each thread to produce memory traces
: 20 5.3 **Stub** routines :
identifies global accesses and adds calls to memory **stub** routines to intercept them. These generate a
www-inf.enst.fr/~research/publications_ec/siegelin/allegro.ps

Dixie: A Retargetable Binary Instrumentation Tool - Manel Fern'andez (1998)   (Correct)
Dixie: A Retargetable Binary **Instrumentation** Tool Manel Fern'andez, Alex Ram'irez, Silvia
projects seeks a a tool that allows flexible **instrumentation** of program binaries to perform com puter
The value generated by the DVM will be collected by **stub** routines (automatically gener ated by JANGO) that

Object Groups: A response to the ORB 2.0 RFI - Isis Distributed (Correct)
way to collect monitoring information from **instrumentation** points in the program and the underlying
CMIP 1990] provide a structured way to define **instrumentation** points in the network that could be
object group support encompassing object services, **stub** and skeleton generators, and high level
www.buva.sowi.uni-bamberg.de/ps-Sammlung/corba/isis_objectgroups.ps.gz

Unknown - (1997) (Correct)
was loaded to failure in the last test. **Instrumentation** Displacements were recorded at 15 locations
was to verify the correct operation of the **instrumentation** and to determine that the load was being
for the McDonnell Douglas AllComposite Wing **Stub** Box Dawn C. Jegley and Harold G. Bush Abstract The
techreports.larc.nasa.gov/pub/techreports/larc/1997/tm/NASA-97-tm110204.ps.Z

In-Kernel Servers on Mach 3.0: Implementation and Performance - Lepreau, Hibler, Ford, Law (Correct)
Mach has failed to optimize the common case. **Instrumentation** of the Mach IPC path on the HewlettPackard
same test as in Tables 1 and 2 without the **instrumentation** code)The cycle count for each test under
to the kernel, the single server, and the RPC **stub** generator. Semantic equivalence, backwards
ftp.cs.utah.edu/pub/inks.ps

Compiler Support for Non-intrusive Monitoring and Debugging.. - Petrov, Stoyen (Correct)
slots for monitoring functions. 4 Automated **instrumentation** Recognizing the need for compiler/tool
an automated and interactive approach to **instrumentation** and monitoring. The general problem in most
established we insert an appropri ate monitoring **stub** for the current idle slot. The **stub** is handled by
cs-ftp.bu.edu/techreports/97-021-ieee-rtss97-wip/petrov.ps

Proxies, Application Interfaces, and Distributed Systems - Amitabh Dave (Correct)
for changing server interfaces, and allow **instrumentation** of server perfor mance. The object model in
to perform all functions normally provided by **stubs** in conventional RPC implementations[11]Last,
a table driven RPC facility in stead of subroutine **stubs**. This mechanism and how it allows proxy objects
choices.cs.uiuc.edu/sefika/iwooos-92.ps.Z

On the design of a 55 GHz Si/SiGe HBT frequency.. - Bruce, Kim.. (Correct)
Tech. Univ. of Munich, Germany, for help with **instrumentation**. The Hewlett Packard is acknowledged for
and output matching to 50 ohms were achieved using **stub** circuits where the **stubs** were terminated in shorts
50 ohms were achieved using **stub** circuits where the **stubs** were terminated in shorts (MIMcapacitors)On
www.signal.uu.se/Publications/ps/doubler7.ps.gz

Documents 21 to 35 Previous 20

Try your query at: Amazon   Barnes & Noble   Google (RI)   Google (Web)   CSB   DBLP

◈IEEE

IEEE Xplore®
RELEASE 1.6

Welcome
United States Patent and Trademark Office

IEEE Xp.
1 Million D
1 Million U
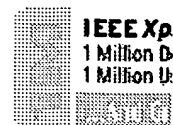
Help   FAQ   Terms   IEEE Peer Review | Quick Links ▾ |

» Search Re

Welcome to IEEE Xplore®

O- Home
O- What Can
   I Access?
O- Log-out

Tables of Contents

O- Journals
   & Magazines
O- Conference
   Proceedings
O- Standards

Search

O- By Author
O- Basic
O- Advanced

Member Services

O- Join IEEE
O- Establish IEEE
   Web Account
O- Access the
   IEEE Member
   Digital Library

Your search matched **11** of **1015452** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

| instrumentation<and>stub | | Search |

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine   **CNF** = Conference   **STD** = Standard

---

1 **Noise measurements of microwave transistors using an uncalibrated mechanical stub tuner and a built-in reverse six-port reflectometer**
Di-Luan Le; Ghannouchi, F.M.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 44 , Issue: 4 , Aug. 1995
Pages:847 - 852

[Abstract]   [PDF Full-Text (396 KB)]   IEEE JNL

---

2 **Modeling, simulation, and measurement considerations of high-speed digital buses**
Novak, I.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 41 , Issue: 6 , Dec. 1992
Pages:921 - 925

[Abstract]   [PDF Full-Text (376 KB)]   IEEE JNL

---

3 **Modeling, simulation, and measurement considerations of high-speed digital buses**
Novak, I.;
Instrumentation and Measurement Technology Conference, 1992. IMTC '92., 9th IEEE , 12-14 May 1992
Pages:147 - 151

[Abstract]   [PDF Full-Text (372 KB)]   IEEE CNF

---

4 **The Level-2 muon trigger at D0**
Fortner, M.; Maciel, A.; Evans, H.; Kothari, B.; Uzunyan, S.;
Nuclear Science, IEEE Transactions on , Volume: 49 , Issue: 4 , Aug. 2002

Pages:1589 - 1592

[Abstract]   [PDF Full-Text (197 KB)]   **IEEE JNL**

---

5 Measurement and simulation of crosstalk reduction by discrete discontinuities along coupled PCB traces
Novak, I.; Eged, B.; Hatvani, L.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 43 , Issue: 2 , Apr 1994
Pages:170 - 175

[Abstract]   [PDF Full-Text (400 KB)]   **IEEE JNL**

---

6 The level-2 muon trigger at D0
Fortner, M.; Maciel, A.; Evans, H.; Kothari, B.; Uzunyan, S.;
Nuclear Science Symposium Conference Record, 2000 IEEE , Volume: 2 , 15-20 Oct. 2000
Pages:12/6 - 12/9 vol.2

[Abstract]   [PDF Full-Text (344 KB)]   **IEEE CNF**

---

7 Reduction and characterization of a drift error in measurement of small antennas using a network analyzer
Ida, I.; Sato, J.; Sekizawa, T.; Yoshimura, H.; Ito, K.;
Antennas, Propagation and EM Theory, 2000. Proceedings. ISAPE 2000. 5th International Symposium on , 15-18 Aug. 2000
Pages:504 - 507

[Abstract]   [PDF Full-Text (288 KB)]   **IEEE CNF**

---

8 A design pattern based approach to generating synchronization adaptors from annotated IDL
Jacobsen, H.-A.; Kramer, B.J.;
Automated Software Engineering, 1998. Proceedings. 13th IEEE International Conference on , 13-16 Oct. 1998
Pages:63 - 72

[Abstract]   [PDF Full-Text (236 KB)]   **IEEE CNF**

---

9 Coupling cavity damper for the ARES
Naito, F.; Akai, K.; Akasaka, N.; Ezura, E.; Kageyama, T.; Mizuno, H.; Nakanishi, H.; Takeuchi, Y.; Yamazaki, Y.; Kobayashi, T.;
Particle Accelerator Conference, 1997. Proceedings of the 1997 , Volume: 3 , 12-16 May 1997
Pages:2977 - 2979 vol.3

[Abstract]   [PDF Full-Text (244 KB)]   **IEEE CNF**

---

10 System control and data acquisition of the two new FWCD RF systems at DIII-D
Harris, T.E.; Allen, J.C.; Cary, W.P.; Ferguson, S.W.; Petty, C.C.; Pinsker, R.I.;
Fusion Engineering, 1995. SOFE '95. 'Seeking a New Energy Era'., 16th IEEE/NPSS Symposium , Volume: 2 , 30 Sept.-5 Oct. 1995

Pages:878 - 880 vol.2

[Abstract]   [PDF Full-Text (320 KB)]   **IEEE CNF**

---

11 **The ICRF computer control and monitoring system**
Mervine, J.; Sichta, P.; Wilson, R.; Giles, D.; Skelly, G.;
Fusion Engineering, 1989. Proceedings., IEEE Thirteenth Symposium on , 2-6 Oct.
1989
Pages:455 - 458 vol.1

[Abstract]   [PDF Full-Text (184 KB)]   **IEEE CNF**

◈IEEE

Membership Publications/Services Standards Conferences Careers/Jobs

IEEE Xplore®
RELEASE 1.6

Welcome
United States Patent and Trademark Office

IEEE Xp.
1 Million D
1 Million U:

Help   FAQ   Terms   IEEE Peer Review | Quick Links ▾ |

» Search Re

Your search matched **11** of **1015452** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or entering a new one in the text box.

| instrument<and>stub | | Search |

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine   **CNF** = Conference   **STD** = Standard

---

1 **Noise measurements of microwave transistors using an uncalibrated mechanical stub tuner and a built-in reverse six-port reflectometer**
Di-Luan Le; Ghannouchi, F.M.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 44 , Issue: 4 , Aug. 1995
Pages:847 - 852

[Abstract]   [PDF Full-Text (396 KB)]   IEEE JNL

---

2 **Modeling, simulation, and measurement considerations of high-speed digital buses**
Novak, I.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 41 , Issue: 6 , Dec. 1992
Pages:921 - 925

[Abstract]   [PDF Full-Text (376 KB)]   IEEE JNL

---

3 **Modeling, simulation, and measurement considerations of high-speed digital buses**
Novak, I.;
Instrumentation and Measurement Technology Conference, 1992. IMTC '92., 9th IEEE , 12-14 May 1992
Pages:147 - 151

[Abstract]   [PDF Full-Text (372 KB)]   IEEE CNF

---

4 **The Level-2 muon trigger at D0**
Fortner, M.; Maciel, A.; Evans, H.; Kothari, B.; Uzunyan, S.;
Nuclear Science, IEEE Transactions on , Volume: 49 , Issue: 4 , Aug. 2002

Pages:1589 - 1592

[Abstract]   [PDF Full-Text (197 KB)]   IEEE JNL

---

5 Measurement and simulation of crosstalk reduction by discrete discontinuities along coupled PCB traces
Novak, I.; Eged, B.; Hatvani, L.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 43 , Issue: 2 , Apr 1994
Pages:170 - 175

[Abstract]   [PDF Full-Text (400 KB)]   IEEE JNL

---

6 The level-2 muon trigger at D0
Fortner, M.; Maciel, A.; Evans, H.; Kothari, B.; Uzunyan, S.;
Nuclear Science Symposium Conference Record, 2000 IEEE , Volume: 2 , 15-20 Oct. 2000
Pages:12/6 - 12/9 vol.2

[Abstract]   [PDF Full-Text (344 KB)]   IEEE CNF

---

7 Reduction and characterization of a drift error in measurement of small antennas using a network analyzer
Ida, I.; Sato, J.; Sekizawa, T.; Yoshimura, H.; Ito, K.;
Antennas, Propagation and EM Theory, 2000. Proceedings. ISAPE 2000. 5th International Symposium on , 15-18 Aug. 2000
Pages:504 - 507

[Abstract]   [PDF Full-Text (288 KB)]   IEEE CNF

---

8 A design pattern based approach to generating synchronization adaptors from annotated IDL
Jacobsen, H.-A.; Kramer, B.J.;
Automated Software Engineering, 1998. Proceedings. 13th IEEE International Conference on , 13-16 Oct. 1998
Pages:63 - 72

[Abstract]   [PDF Full-Text (236 KB)]   IEEE CNF

---

9 Coupling cavity damper for the ARES
Naito, F.; Akai, K.; Akasaka, N.; Ezura, E.; Kageyama, T.; Mizuno, H.; Nakanishi, H.; Takeuchi, Y.; Yamazaki, Y.; Kobayashi, T.;
Particle Accelerator Conference, 1997. Proceedings of the 1997 , Volume: 3 , 12-16 May 1997
Pages:2977 - 2979 vol.3

[Abstract]   [PDF Full-Text (244 KB)]   IEEE CNF

---

10 System control and data acquisition of the two new FWCD RF systems at DIII-D
Harris, T.E.; Allen, J.C.; Cary, W.P.; Ferguson, S.W.; Petty, C.C.; Pinsker, R.I.;
Fusion Engineering, 1995. SOFE '95. 'Seeking a New Energy Era'., 16th IEEE/NPSS Symposium , Volume: 2 , 30 Sept.-5 Oct. 1995

Pages:878 - 880 vol.2

[Abstract]   [PDF Full-Text (320 KB)]   IEEE CNF

11 The ICRF computer control and monitoring system

Mervine, J.; Sichta, P.; Wilson, R.; Giles, D.; Skelly, G.;
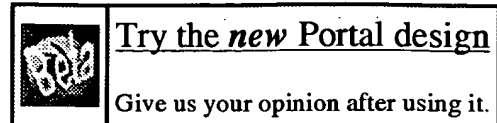Fusion Engineering, 1989. Proceedings., IEEE Thirteenth Symposium on , 2-6 Oct. 1989
Pages:455 - 458 vol.1

[Abstract]   [PDF Full-Text (184 KB)]   IEEE CNF

PORTAL
THE ACM DIGITAL LIBRARY

Try the *new* Portal design

Give us your opinion after using it.

Search Results

Search Results for: **[instrument<AND>(((generating stub)))]**
Found **18** of **129,310 searched.**

Search within Results

> Advanced Search   > Search Help/Tips

Sort by: Title   Publication   Publication Date   Score   ❦ Binder

---

Results 1 - 18 of 18   short listing

---

1 Technical papers: software design: DADO: enhancing middleware to support   84%
crosscutting features in distributed, heterogeneous systems
Eric Wohlstadter , Stoney Jackson , Premkumar Devanbu
Proceedings of the 25th international conference on Software engineering May 2003
> Some "non-" or "extra-functional" features, such as reliability, security, and tracing, defy
> modularization mechanisms in programming languages. This makes such features hard to
> design, implement, and maintain. Implementing such features within a single platform,
> using a single language, is hard enough. With distributed, heterogeneous (DH) systems,
> these features induce complex implementations which cross-cut different languages, OSs,
> and hardware platforms, while still needing to share data and ...

2 Evolving RPC for active storage   80%
Muthian Sivathanu , Andrea C. Arpaci-Dusseau , Remzi H. Arpaci-Dusseau
**Tenth international conference on architectural support for programming languages
and operating systems on Proceedings of the 10th international conference on
architectural support for programming languages and operating systems (ASPLOS-X)**
October 2002
Volume 37 , 36 , 30 Issue 10 , 5 , 5
> We introduce Scriptable RPC (SRPC), an RPC-based framework that enables distributed
> system services to take advantage of active components. Technology trends point to a world
> where each component in a system (whether disk, network interface, or memory) has
> substantial computational capabilities; however, traditional methods of building distributed
> services are not designed to take advantage of these new architectures, mandating
> wholesale change of the software base to exploit more powerful hardw ...

3 Why IDLs are Not Ideal   77%
A. Kaplan , J. Ridgway , J. C. Wileden
Proceedings of the 9th International Workshop on Software Specification and Design
April 1998

4 Lightweight remote procedure call   77%

Brian N. Bershad , Thomas E. Anderson , Edward D. Lazowska , Henry M. Levy
ACM Transactions on Computer Systems (TOCS) February 1990
Volume 8 Issue 1

Lightweight Remote Procedure Call (LRPC) is a communication facility designed and optimized for communication between protection domains on the same machine. In contemporary small-kernel operating systems, existing RPC systems incur an unnecessarily high cost when used for the type of communication that predominates—between protection domains on the same machine. This cost leads system designers to coalesce weakly related subsystems into the same protection domain, trading safety for ...

5 Lightweight remote procedure call   77%

B. Bershad , T. Anderson , E. Lazowska , H. Levy
ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles November 1989
Volume 23 Issue 5

Lightweight Remote Procedure Call (LRPC) is a communication facility designed and optimized for communication between protection domains on the same machine. In contemporary small-kernel operating systems, existing RPC systems incur an unnecessarily high cost when used for the type of communication that predominates — between protection domains on the same machine. This cost leads system designers to coalesce weakly-related subsystems into the same protection domain, tradin ...

6 Performance and scalability of EJB applications   77%

Emmanuel Cecchet , Julie Marguerite , Willy Zwaenepoel
ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications November 2002
Volume 37 Issue 11

We investigate the combined effect of application implementation method, container design, and efficiency of communication layers on the performance scalability of J2EE application servers by detailed measurement and profiling of an auction site server.We have implemented five versions of the auction site. The first version uses stateless session beans, making only minimal use of the services provided by the Enterprise JavaBeans (EJB) container. Two versions use entity beans, one with container- ...

7 Recompilation for debugging support in a JIT-compiler   77%

Mustafa M. Tikir , Jeffrey K. Hollingsworth , Guei-Yuan Lueh
ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering November 2002
Volume 28 Issue 1

A static Java compiler converts Java source code into a verifiably secure and compact architecture-neutral intermediate format, called Java *byte codes*. The Java byte codes can be either interpreted by a Java Virtual Machine or translated into native code by Java Just-In-Time compilers. Static Java compilers embed debug information in the Java class files to be used by the source level debuggers. However, the debug information is generated for architecture independent byte codes and most o ...

8 Evaluating the performance limitations of MPMD communication   77%

Chi-Chao Chang , Grzegorz Czajkowski , Thorsten von Eicken , Carl Kesselman
Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM) November 1997

The MPMD approach for parallel computing is attractive for programmers who seek fast development cycles, high code re-use, and modular programming, or whose applications exhibit irregular computation loads and communication patterns. RPC is widely adopted as

the communication abstraction for crossing address space boundaries. However, the communication overheads of existing RPC-based systems are usually an order of magnitude higher than those found in highly tuned SPMD systems. This problem has ...

9 Profile-guided code compression   77%

Saumya Debray , William Evans
ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation May 2002
Volume 37 Issue 5

> As computers are increasingly used in contexts where the amount of available memory is limited, it becomes important to devise techniques that reduce the memory footprint of application programs while leaving them in an executable form. This paper describes an approach to applying data compression techniques to reduce the size of infrequently executed portions of a program. The compressed code is decompressed dynamically (via software) if needed, prior to execution. The use of data compression t ...

10 Efficient wire formats for high performance computing   77%

Fabian E. Bustamante , Greg Eisenhauer , Karsten Schwan , Patrick Widener
Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM) November 2000

> High performance computing is being increasingly utilized in non-traditional circumstances where it must interoperate with other applications. For example, online visualization is being used to monitor the progress of applications, and real-world sensors are used as inputs to simulations. Whenever these situations arise, there is a question of what communications infrastructure should be used to link the different components. Traditional HPC-style communications systems such as MPI offer r ...

11 SAFKASI: a security mechanism for language-based systems   77%

Dan S. Wallach , Andrew W. Appel , Edward W. Felten
ACM Transactions on Software Engineering and Methodology (TOSEM) October 2000
Volume 9 Issue 4

> In order to run untrusted code in the same process as trusted code, there must be a mechanism to allow dangerous calls to determine if their caller is authorized to exercise the privilege of using the dangerous routine. Java systems have adopted a technique called stack inspection to address this concern. But its original definition, in terms of searching stack frames, had an unclear relationship to the actual achievement of security, overconstrained the implementation of a Java system, lim ...

12 Practicing JUDO: Java under dynamic optimizations   77%

Michał Cierniak , Guei-Yuan Lueh , James M. Stichnoth
ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation May 2000
Volume 35 Issue 5

> A high-performance implementation of a Java Virtual Machine (JVM) consists of efficient implementation of Just-In-Time (JIT) compilation, exception handling, synchronization mechanism, and garbage collection (GC). These components are tightly coupled to achieve high performance. In this paper, we present some static anddynamic techniques implemented in the JIT compilation and exception handling of the Microprocessor Research Lab Virtual Machine (MRL VM), ...

13 Hybrid domain-specific kits for a flexible software factory   77%

Martin L. Griss , Kevin D. Wentzel
Proceedings of the 1994 ACM symposium on Applied computing April 1994

Results 1 - 18 of 18   short listing

PORTAL
THE ACM DIGITAL LIBRARY

Search Results

Search Results for: **[compiling<AND>((instrumentation<AND>(((generating stub)))))]**
Found **15** of **129,310 searched.**

Search within Results

> Advanced Search   > Search Help/Tips

Sort by: Title   Publication   Publication Date   Score   ❤ Binder

Results 1 - 15 of 15   short listing

1 Technical papers: software design: DADO: enhancing middleware to support 84%
crosscutting features in distributed, heterogeneous systems
Eric Wohlstadter , Stoney Jackson , Premkumar Devanbu
Proceedings of the 25th international conference on Software engineering May 2003
Some "non-" or "extra-functional" features, such as reliability, security, and tracing, defy
modularization mechanisms in programming languages. This makes such features hard to
design, implement, and maintain. Implementing such features within a single platform,
using a single language, is hard enough. With distributed, heterogeneous (DH) systems,
these features induce complex implementations which cross-cut different languages, OSs,
and hardware platforms, while still needing to share data and ...

2 Lightweight remote procedure call   77%
Brian N. Bershad , Thomas E. Anderson , Edward D. Lazowska , Henry M. Levy
ACM Transactions on Computer Systems (TOCS) February 1990
Volume 8 Issue 1
Lightweight Remote Procedure Call (LRPC) is a communication facility designed and
optimized for communication between protection domains on the same machine. In
contemporary small-kernel operating systems, existing RPC systems incur an unnecessarily
high cost when used for the type of communication that predominates—between protection
domains on the same machine. This cost leads system designers to coalesce weakly related
subsystems into the same protection domain, trading safety for ...

3 Lightweight remote procedure call   77%
B. Bershad , T. Anderson , E. Lazowska , H. Levy
**ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium
on Operating systems principles** November 1989
Volume 23 Issue 5
Lightweight Remote Procedure Call (LRPC) is a communication facility designed and
optimized for communication between protection domains on the same machine. In
contemporary small-kernel operating systems, existing RPC systems incur an unnecessarily
high cost when used for the type of communication that predominates — between protection

domains on the same machine. This cost leads system designers to coalesce weakly-related subsystems into the same protection domain, tradin ...

4 Evolving RPC for active storage   77%

Muthian Sivathanu , Andrea C. Arpaci-Dusseau , Remzi H. Arpaci-Dusseau
Tenth international conference on architectural support for programming languages and operating systems on Proceedings of the 10th international conference on architectural support for programming languages and operating systems (ASPLOS-X)
October 2002
Volume 37 , 36 , 30 Issue 10 , 5 , 5

> We introduce Scriptable RPC (SRPC), an RPC-based framework that enables distributed system services to take advantage of active components. Technology trends point to a world where each component in a system (whether disk, network interface, or memory) has substantial computational capabilities; however, traditional methods of building distributed services are not designed to take advantage of these new architectures, mandating wholesale change of the software base to exploit more powerful hardw ...

5 Performance and scalability of EJB applications   77%

Emmanuel Cecchet , Julie Marguerite , Willy Zwaenepoel
ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications November 2002
Volume 37 Issue 11

> We investigate the combined effect of application implementation method, container design, and efficiency of communication layers on the performance scalability of J2EE application servers by detailed measurement and profiling of an auction site server.We have implemented five versions of the auction site. The first version uses stateless session beans, making only minimal use of the services provided by the Enterprise JavaBeans (EJB) container. Two versions use entity beans, one with container- ...

6 Recompilation for debugging support in a JIT-compiler   77%

Mustafa M. Tikir , Jeffrey K. Hollingsworth , Guei-Yuan Lueh
ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering November 2002
Volume 28 Issue 1

> A static Java compiler converts Java source code into a verifiably secure and compact architecture-neutral intermediate format, called Java *byte codes*. The Java byte codes can be either interpreted by a Java Virtual Machine or translated into native code by Java Just-In-Time compilers. Static Java compilers embed debug information in the Java class files to be used by the source level debuggers. However, the debug information is generated for architecture independent byte codes and most o ...

7 Evaluating the performance limitations of MPMD communication   77%

Chi-Chao Chang , Grzegorz Czajkowski , Thorsten von Eicken , Carl Kesselman
Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM) November 1997

> The MPMD approach for parallel computing is attractive for programmers who seek fast development cycles, high code re-use, and modular programming, or whose applications exhibit irregular computation loads and communication patterns. RPC is widely adopted as the communication abstraction for crossing address space boundaries. However, the communication overheads of existing RPC-based systems are usually an order of magnitude higher than those found in highly tuned SPMD systems. This problem has ...

8 Profile-guided code compression   77%

Saumya Debray , William Evans
**ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on
Programming language design and implementation** May 2002
Volume 37 Issue 5

As computers are increasingly used in contexts where the amount of available memory is
limited, it becomes important to devise techniques that reduce the memory footprint of
application programs while leaving them in an executable form. This paper describes an
approach to applying data compression techniques to reduce the size of infrequently
executed portions of a program. The compressed code is decompressed dynamically (via
software) if needed, prior to execution. The use of data compression t ...

9 Efficient wire formats for high performance computing   77%

Fabian E. Bustamante , Greg Eisenhauer , Karsten Schwan , Patrick Widener
Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM) November
2000

High performance computing is being increasingly utilized in non-traditional circumstances
where it must interoperate with other applications. For example, online visualization is being
used to monitor the progress of applications, and real-world sensors are used as inputs to
simulations. Whenever these situations arise, there is a question of what communications
infrastructure should be used to link the different components. Traditional HPC-style
communications systems such as MPI offer r ...

10 SAFKASI: a security mechanism for language-based systems   77%

Dan S. Wallach , Andrew W. Appel , Edward W. Felten
ACM Transactions on Software Engineering and Methodology (TOSEM) October 2000
Volume 9 Issue 4

In order to run untrusted code in the same process as trusted code, there must be a
mechanism to allow dangerous calls to determine if their caller is authorized to exercise the
privilege of using the dangerous routine. Java systems have adopted a technique called
stack inspection to address this concern. But its original definition, in terms of searching
stack frames, had an unclear relationship to the actual achievement of security,
overconstrained the implementation of a Java system, lim ...

11 Practicing JUDO: Java under dynamic optimizations   77%

Michał Cierniak , Guei-Yuan Lueh , James M. Stichnoth
**ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on
Programming language design and implementation** May 2000
Volume 35 Issue 5

A high-performance implementation of a Java Virtual Machine (JVM) consists of efficient
implementation of Just-In-Time (JIT) compilation, exception handling, synchronization
mechanism, and garbage collection (GC). These components are tightly coupled to achieve
high performance. In this paper, we present some static anddynamic techniques
implemented in the JIT compilation and exception handling of the Microprocessor Research
Lab Virtual Machine (MRL VM), ...

12 Hybrid domain-specific kits for a flexible software factory   77%

Martin L. Griss , Kevin D. Wentzel
Proceedings of the 1994 ACM symposium on Applied computing April 1994

13 The runtime creation of code for printing simulation output   77%

John H. Reynolds

14 Flick: a flexible, optimizing IDL compiler 77%

Eric Eide , Kevin Frei , Bryan Ford , Jay Lepreau , Gary Lindstrom
ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation May 1997
Volume 32 Issue 5

> An interface definition language (IDL) is a nontraditional language for describing interfaces between software components. IDL compilers generate "stubs" that provide separate communicating processes with the abstraction of local object invocation or procedure call. High-quality stub generation is essential for applications to benefit from component-based designs, whether the components reside on a single computer or on multiple networked hosts. Typical IDL compilers, ...

15 The POLYLITH software bus 77%

James M. Purtilo
ACM Transactions on Programming Languages and Systems (TOPLAS) January 1994
Volume 16 Issue 1

> We describe a system called POLYLITH that helps programmers prepare and interconnect mixed-language software components for execution in heterogeneous environments. POLYLITH's principal benefit is that programmers are free to implement functional requirements separately from their treatment of interfacing requirements; this means that once an application has been developed for use in one execution environment (such as a distributed network) it can be adapted for reuse in other environments ...

Results 1 - 15 of 15   short listing